

This listing of claims will replace all prior versions, and listings, of claims in the application.

LISTING OF CLAIMS:

Claims 1-20 (Cancelled).

21. (Currently Amended) A method of managing a cluster of networked resources using rule-based constraints in a scalable clustering environment, the method comprising the steps of:

building a globally optimal ~~configurations~~ configuration of said cluster of resources, wherein each of the resources has an availability and quality of service, and the availabilities and quality of services of the resources are determined by dependencies among the resources, user preferences, constraints on the resources, events, and network policies,

bringing said cluster of resources on-line in a systematic manner, ~~given a set of resources, resource groups,~~ current states of said resources and resource groups, and said dependencies, preferences, constraints, events, and policies,

determining dynamic dependencies of and configuration information about said cluster of resources, including determining said dependencies and configuration information (i) at cluster initialization and (ii) dynamically during cluster operation,

supporting startup and shutdown of said cluster of resources according to current policies, and system events,

separating [[the]] said dependencies, preferences, constraints, events, and policies into (i) a first rules based and-objectives group and (ii) a second, dynamically changing eluster events and policies based group, and

~~separating the networked resources, resource groups, and cluster configurations into a first, static group, and a second, dynamically changing group, and~~

taking a snapshot of the combining said first and second groups in a systematic manner only when needed to build the said optimal configuration.

22. (Previously Presented) A method according to Claim 21, comprising the further steps of:

continuously monitoring cluster-wide events and comparing the current cluster state with a desired state, and whenever there is a discrepancy between said current and desired states, realigning the cluster resources, including the step of issuing commands to the cluster resources to bring about the realigning;

providing a group of cluster services, including:

- i) a persistent cluster registry to store and retrieve the configuration of the cluster resources,
- ii) topology services for detecting node and communication adapter failures,
- iii) messaging for selected communications between a central resource and all other resources, and
- iv) a group services facility for electing one of the resources as the central resource at cluster initialization and whenever an existing central resource is unable to provide the services thereof,

delivering events to a coordinator, said coordinator combining said events with said rules and objectives to arrive at a response to said events;

translating the response into commands to the resources, each of the commands containing all the state needed for execution of the command by a manager of one of the resources, including the step of issuing the commands in a partial order given by said dependencies; and

not sending out a new command until the central resource is aware of a positive outcome of the commands that the execution of said new command depends on.

23. (Previously Presented) A method according to Claim 22, wherein:

said coordinator ensures that globally-optimal solutions get deployed in the cluster in response to events in the cluster; and

all events and command feedback are directed to said coordinator.

24. (Previously Presented) A method according to Claim 21, comprising the further steps of:

providing an optimizer module for computing a globally optimal solution based on said constraints and to current state of the cluster;

using the optimizer for recomputing the globally optimal solution whenever an objective value of a deployed solution is below a certain value as compared to a proposed solution, including the step of feeding back to the optimizer an artificially generated event that forces the optimizer to recompute the global solution;

providing the optimizer with a snapshot of the current state of the cluster;

wherein the step of using the optimizer for recomputing the globally optimal solution includes the step of said optimizer, given said snapshot, proposing an approximately optimal cluster configuration that takes into account said current state of the cluster and long-term objectives defined for the cluster.

25. (Currently Amended) A system for managing a cluster of networked resources using rule-based constraints in a scalable clustering environment, comprising apparatus for:

building a globally optimal ~~configurations~~ configuration of said cluster of resources, wherein each of the resources has an availability and quality of service, and the availabilities and quality of services of the resources are determined by dependencies among the resources, user preferences, constraints on the resources, events, and network policies,

bringing said cluster of resources on-line in a systematic manner, given ~~a set of resources, resource groups,~~ current states of said resources and resource groups, and said dependencies, preferences, constraints, events, and policies,

determining dynamic dependencies of configuration information about said cluster of resources, including determining said dependencies and configuration information (i) at cluster initialization and (ii) dynamically during cluster operation,

supporting a startup and shutdown of said cluster of resources according to current policies, and system events,

separating the said dependencies, preferences, constraints, events, and policies into (i) a first rules and objectives based group and (ii) a second, dynamically changing ~~cluster events and policies based~~ group, and

~~separating the networked resources, resource groups, and cluster configurations into a first, static group and a second dynamically changing group, and~~

~~taking a snapshot of the~~ combining said first and second groups in a systematic manner only when needed to build the said optimal configuration.

26. (Currently Amended) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps of managing a cluster of networked resources using rule-based constraints in a scalable clustering environment, said method steps comprising:

building a globally optimal ~~configurations~~ configuration of said cluster of resources, wherein each of the resources has an availability and quality of service, and the availabilities and quality of services of the resources are determined by dependencies among the resources, user preferences, constraints on the resources, events, and network policies,

bringing said cluster of resources on-line in a systematic manner, ~~given a set of resources, resource groups,~~ current states of said resources and resource groups, and said dependencies, preferences, constraints, events, and policies,

determining dynamic dependencies of configuration information about said cluster of resources, including determining said dependencies and configuration information (i) at cluster initialization and (ii) dynamically during cluster operation,

supporting startup and shutdown of said cluster of resources according to current policies, and system events,

separating [[the]] said dependencies, preferences, constraints, events, and policies into (i) a first rules ~~and objectives~~ based group and (ii) a second, dynamically changing ~~cluster~~ events and ~~policies~~ based group, and

~~separating the networked resources, resource groups, and cluster configurations into a first, static group, and a second dynamically changing group, and~~

~~taking a snapshot of the~~ combining said first and second groups only when needed to build the said optimal configuration.

27. (New) A method according to Claim 24, wherein:

the providing step includes the step of providing a preprocessor module and a postprocessor module;

the preprocessor module includes a preprocessor entry queue, the optimizer module includes an optimizer input queue, and the postprocessor module includes a postprocessor input queue;

a decision to do a resource reallocation results in the creation of a preprocessor task that is deposited in the entry queue of the preprocessor module; and

wherein said task is an object having an entry method that, when invoked, results in the task being executed, and execution of the task results in either a postprocessor task being deposited in the postprocessor input queue, an optimizer task being deposited in the optimizer input queue, or both;

both the post processor task and the optimizer task are scheduled by an invocation of the entry method associated with the individual tasks; and

execution of the optimizer task results in a postprocessor task being deposited in the postprocessor input queue.